Döcu Content

Google Worksheets

Java Script Library

Console Application

System Requirements:
Download Domino Designer **8.5.3** Environment (DDE)
http://www.ibm.com/developerworks/downloads/ls/dominodesigner/

Download /Make available Google Spreadsheets API
https://developers.google.com/google-apps/spreadsheets/
https://developers.google.com/google-apps/documents-list/

Introduction:
In this tutorial, we are again using a Java console program to list the name of a certain Worksheet within the current Spreadsheet, housed on Google Drive. We will be reading publically available document(s), previously created manually... We will be programming via Domino Desginer Environement (DDE), all prior JAR configurations will work for this excercise.

Disclaimer:
Information contained in the following is presented as is. This tutorial assumes you have basic Lotus Notes Configuration, Programming knowledge, and are familiar with Google APIs.

Döcu Content JavaAgent (Script Library)

At this point, we assume you have gone through the prior tutorial, reading existing Spreadsheet. Copy and paste code below into DDE and submit a new Worksheet. Be sure to Reference the JAR file Script Library in your JavaAgent program, run code to list Worksheet(s) info to console window....

Related Info:
https://www.youtube.com/watch?v=CTnUOLcOfgI&list=UUSImDTpK0oe7QrPsYOE4nww

Copy and Paste Page Design code

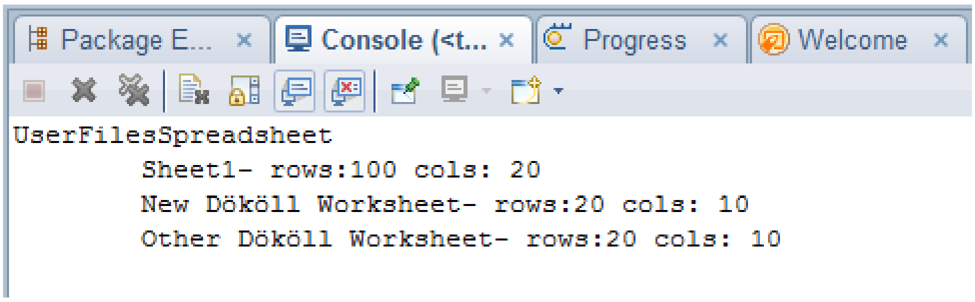
More than likely, with the proper JAR files in place, the program compiles, run as Java program and view results in DDE Console window and your Google Drive.

NewGoogleDriveWorkSheetJavaAgent.java;

```
/**
 * Created from copy: 2014.05.13.7.34.PM
 * GoogleSpreadsheetsWorkSheetJavaAgent | NewGoogleDriveWorkSheetJavaAgent.java
 * List Worksheets in specific Spreadsheet, housed on Google Drive
 */

//...
//Google imports
import com.google.gdata.client.spreadsheet.*;
import com.google.gdata.data.PlainTextConstruct;
import com.google.gdata.data.spreadsheet.*;
import com.google.gdata.util.*;
//...
//Java imports
import java.io.IOException;
import java.net.*;
import java.util.*;

/**
 * @author Dököll Solutions, Inc.
 * @version 2014.05.13.7.34.PM
 *

public class NewGoogleDriveWorkSheetJavaAgent {
  public static void main(String[] args)
      throws AuthenticationException, MalformedURLException, IOException,
ServiceException {

        //TODO: see about connecting automatically...
        //existing Google Drive login creds...
    String USERNAME = "yourgoogleaccount@gmail.com";
    String PASSWORD = "youraccountpassword";

    SpreadsheetService service =
        new SpreadsheetService("MySpreadsheetIntegration");
    service.setUserCredentials(USERNAME, PASSWORD);

        // TODO: Authorize the service object for a specific user (see other
sections)

    // Define the URL to request.  This should never change.
    URL SPREADSHEET_FEED_URL = new URL(
        "https://spreadsheets.google.com/feeds/spreadsheets/private/full");

    // Make a request to the API and get all spreadsheets.
    SpreadsheetFeed feed = service.getFeed(SPREADSHEET_FEED_URL,
        SpreadsheetFeed.class);
    List<SpreadsheetEntry> spreadsheets = feed.getEntries();

    if (spreadsheets.size() == 0) {
      // TODO: There were no spreadsheets, act accordingly.
    }

    // TODO: Choose a spreadsheet more intelligently based on your
    // app's needs.
    SpreadsheetEntry spreadsheet = spreadsheets.get(0);
    System.out.println(spreadsheet.getTitle().getPlainText());

    // Create a local representation of the new worksheet.
    WorksheetEntry worksheet = new WorksheetEntry();
    worksheet.setTitle(new PlainTextConstruct("Other Dököll Worksheet"));
    worksheet.setColCount(10);
    worksheet.setRowCount(20);

    // Send the local representation of the worksheet to the API for
    // creation.  The URL to use here is the worksheet feed URL of our
    // spreadsheet.
    URL worksheetFeedUrl = spreadsheet.getWorksheetFeedUrl();
    service.insert(worksheetFeedUrl, worksheet);

  }
}
```

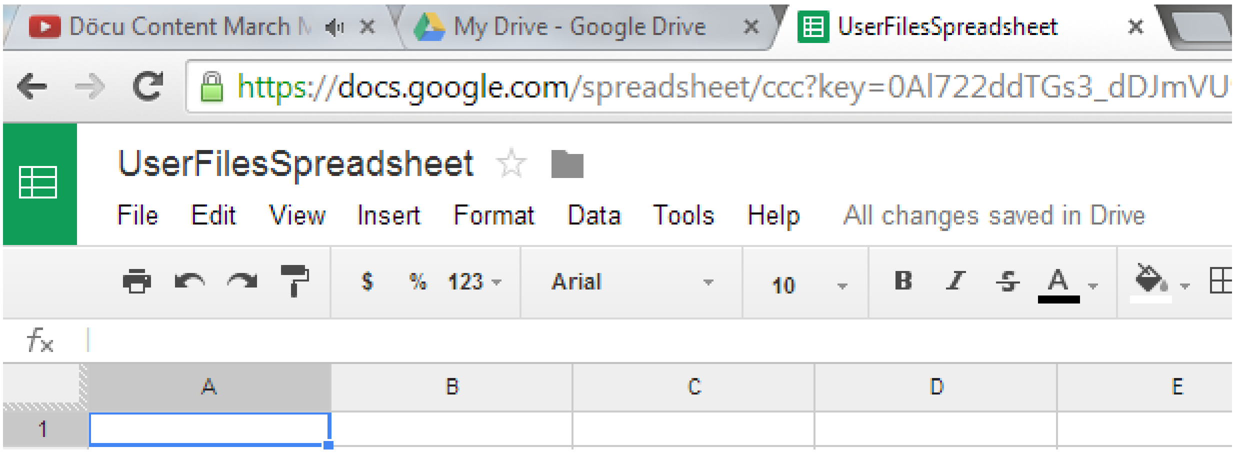
Döcu Content Console View (Console/Drive)

Full page design: Here is what your results might look like when program runs...

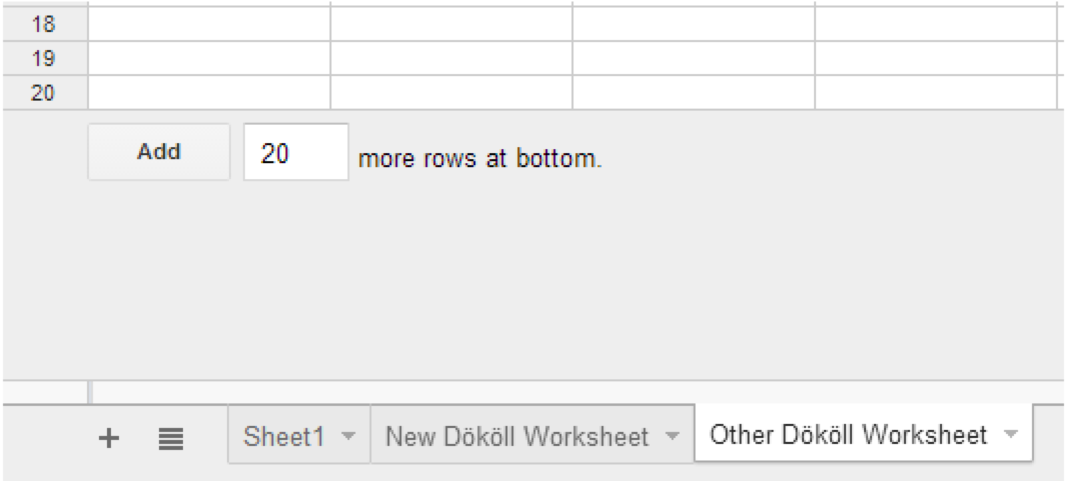


```
UserFilesSpreadsheet
        Sheet1- rows:100 cols: 20
        New Dököll Worksheet- rows:20 cols: 10
        Other Dököll Worksheet- rows:20 cols: 10
```

top part of the document



bottom part of the document



TIP: Extract part of the Spreadsheet listing code that loops through and lists the worksheet(s) by name into Console Window, plug it into the above, run it...

Conclusion:

You can now add Worksheets to existing Spreadsheet, publically available on Google's Infrastructure.

Questions, comments, please post a brief message on our Contact form on the main site.

Thank you for coming...

Version:2014.05.13.7.57.PM